DETAILED ACTION

1. This Office action is in response to the amendment filed on September 28, 2009.

2. **Claims 1, 4-9, 12-19, and 21-26** are pending.

3. **Claims 1, 6-9, 12, and 15-19** have been amended.

4. **Claims 2, 3, 10, 11, 20, and 27** have been canceled.

5. Applicant's amendments to the claims fail to fully address the objections to Claims 19 and 21-26 due to improper antecedent basis. Accordingly, these objections are maintained and further explained hereinafter.

Response to Amendment

Claim Objections

6. **Claims 15-19 and 21-26** are objected to because of the following informalities:

- **Claims 15 and 16** recite the limitation "said coding module." Applicant is advised to change this limitation to read "said located coding module" for the purpose of providing it with proper explicit antecedent basis.
- **Claims 17 and 18** depend on Claim 15 and, therefore, suffer the same deficiency as Claim 15.
- **Claims 17 and 18** recite the limitation "instructions." Applicant is advised to change this limitation to read "computer executable instructions" for the purpose of keeping the claim language consistent throughout the claims.

- **Claim 19** recites the limitation "said grid." Applicant is advised to change this limitation to read "said computational grid" for the purpose of providing it with proper explicit antecedent basis.
- **Claims 21-26** depend on Claim 19 and, therefore, suffer the same deficiency as Claim 19.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. **Claims 1, 4-9, 12-19, and 21-26** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1 and 12 recite the limitation "the computational grid." There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "a computational grid" for the purpose of further examination.

Claims 4-9 depend on Claim 1 and, therefore, suffer the same deficiency as Claim 1.

Claims 13-18 depend on Claim 12 and, therefore, suffer the same deficiency as Claim 12.

Claim 19 recites the limitation "a web service coupled to the web service." The claim is rendered indefinite because it is unclear to the Examiner how a web service can be coupled to itself. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "a web service coupled to the computational grid" for the purpose of further examination.

Claims 21-26 depend on Claim 19 and, therefore, suffer the same deficiency as Claim 19.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. **Claims 1, 4-9, 12-19, and 21-26** are rejected under 35 U.S.C. 103(a) as being unpatentable over **US 2004/0088688 (hereinafter "Hejlsberg")** in view of **US 7,185,046 (hereinafter "Ferstl")**.

As per **Claim 1**, Hejlsberg discloses:

- receiving from an author over a computer communications network a description of a computing application in a web service executing in memory by a processor in a computer *(see Figures 1 and 2; Paragraph [0006], "... a file, such as a database mapping description or declaration, is authored by a user or a design tool in a particular data language in which a format can be defined, such as XML. Such an exemplary file is referred to as a blueprint ..."; Paragraph [0017], "Blueprints allow the ASP.NET markup-and-code paradigm to be extended to other domains such as user interfaces, database mapping, web services, and compiled extensible stylesheet language (XSL) transforms."; Paragraph [0087], "The present invention can be applied to a wide variety of technologies, such as ... web services ...")*;

- parsing said description in said web service to identify object parameters for said computing application *(see Paragraph [0035], "Upon receiving the blueprint 200, the blueprint translator 210 parses the blueprint (using, e.g., an XML parser) ..."; Paragraph [0047], "In addition, the framework defines a file extension, .dbml, and includes a blueprint translator that can translate .dbml files containing XML-formatted mapping descriptions into source code that targets the framework. An exemplary .dbml blueprint is generated by a user or a design tool and is shown below.")*;

- supplying said description to a node *(see Paragraph [0035], "... provides the parsed blueprint to a Document Object Model (DOM) for further processing. The output of the DOM is provided to a semantic analyzer and code generator. Source code 220 is thereby generated in accordance with predetermined schemas, patterns, and/or hierarchical rules, for example.")*;

- applying said description to a located coding module to generate at least one output object corresponding to the identified object parameters *(see Paragraph [0035], "... provides the parsed blueprint to a Document Object Model (DOM) for further processing. The output of the DOM is provided to a semantic analyzer and code generator. Source code 220 is thereby generated in accordance with predetermined schemas, patterns, and/or hierarchical rules, for example."; Paragraph [0058], "... a blueprint translator can use the CodeDOM (an object model for abstract syntax trees and code generation provided in the System.CodeDom namespace) to generate source code in a language-neutral fashion.")*; and

- returning said at least one output object to the author over the computer communications network *(see Paragraph [0035], "The source code 220 may access or point to a supporting framework or class library 230.")*.

However, Hejlsberg does not disclose:

- locating a coding module corresponding to at least one of the object parameters within a node contained within a computational grid coupled to the web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding modules.

Ferstl discloses:

- locating a coding module corresponding to at least one of object parameters within a node contained within a computational grid coupled to a web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding

modules *(see Column 1: 52-59, "A computing grid is a hardware and software infrastructure serving to handle computing jobs submitted by a user. The computing grid may interconnect distributed computers, storage devices, mobile devices, instruments, sensors, data bases and/or software applications. Generally a computing grid may comprise virtually any kind of computing device and includes a grid infrastructure to handle the distribution of computing jobs." and 65-67 to Column 2: 1-8, "Upon receiving an instruction to distribute a computing job the grid infrastructure selects a suitable computing device and transfers the computing job to the selected computing device." and "Accordingly, a user or application at a client device may issue an instruction to execute a computing job towards the grid infrastructure which in turn selects a suitable processing element and the processing results are ultimately returned to the client."; Column 12: 25-31, "In an example, the selection section obtains the selection information and accesses data specifying corresponding features of a plurality of job handlers, for example, in a configuration file specifying features of a plurality of job handlers available. Then, the selection section may identify a suitable job handler matching the selection information in association with the job request.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Ferstl into the teaching of Hejlsberg to modify Hejlsberg's invention to include locating a coding module corresponding to at least one of the object parameters within a node contained within a computational grid coupled to the web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding modules. The modification would be obvious because one of ordinary skill in

the art would be motivated to utilize a variety of computing devices and/or software applications to quickly solve a single computing task *(see Ferstl – Column 1: 46-50)*.

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and Hejlsberg further discloses:

- wherein said description is generated using Object Meta Language (OML) *(see Paragraph [0006], "... a file, such as a database mapping description or declaration, is authored by a user or a design tool in a particular data language in which a format can be defined, such as XML. Such an exemplary file is referred to as a blueprint ...")*.

As per **Claim 5**, the rejection of **Claim 4** is incorporated; and Hejlsberg further discloses:

- wherein said OML is an eXtensible Markup Language (XML) dialect *(see Paragraph [0006], "... a file, such as a database mapping description or declaration, is authored by a user or a design tool in a particular data language in which a format can be defined, such as XML. Such an exemplary file is referred to as a blueprint ...")*.

As per **Claim 6**, the rejection of **Claim 1** is incorporated; and Hejlsberg further discloses:

- wherein said located coding module is an XML template *(see Paragraph [0047], "In addition, the framework defines a file extension, .dbml, and includes a blueprint translator that can translate .dbml files containing XML-formatted mapping descriptions into source code that targets the framework.")*.

As per **Claim 7**, the rejection of **Claim 1** is incorporated; and Hejlsberg further discloses:

- wherein said located coding module is an eXtensible Stylesheet Language (XSL) style sheet *(see Paragraph [0017], "Blueprints allow the ASP.NET markup-and-code paradigm to be extended to other domains such as user interfaces, database mapping, web services, and compiled extensible stylesheet language (XSL) transforms.").*

As per **Claim 8**, the rejection of **Claim 7** is incorporated; and Hejlsberg further discloses:

- parsing said description to locate at least one variable *(see Paragraph [0048], "... mapping the Customers table in the database to a Customer class in the Northwind namespace. Further details of the mapping include the CustomerID column that maps to an Id property, the ContactName column that maps to a Name property, etc.")*; and

- substituting said at least one variable with at least one replacement variable, wherein said at least one replacement variable is the result of an XML/XSL transform *(see Paragraph [0048], "... the blueprint calls for an Orders collection to be generated in the Customer class based on the relation between the Customer and Order classes described in the <relation> element."; Paragraph [0050], "A blueprint like the one set forth above would typically be generated by a database design tool, but it could also be authored manually or created by an XML transformation.").*

As per **Claim 9**, the rejection of **Claim 6** is incorporated; and Hejlsberg further discloses:

- parsing said description to locate at least one variable *(see Paragraph [0048], "... mapping the Customers table in the database to a Customer class in the Northwind namespace.*

Further details of the mapping include the CustomerID column that maps to an Id property, the ContactName column that maps to a Name property, etc."); and

- substituting said at least one variable with at least one replacement variable, wherein said at least one replacement variable is stored in said XML template *(see Paragraph [0048], "... the blueprint calls for an Orders collection to be generated in the Customer class based on the relation between the Customer and Order classes described in the <relation> element."; Paragraph [0050], "A blueprint like the one set forth above would typically be generated by a database design tool, but it could also be authored manually or created by an XML transformation.")*.

Claims 12-18 are computer program product claims corresponding to the method claims above (Claims 1 and 4-9) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1 and 4-9.

As per **Claim 19**, Hejlsberg discloses:

- a web service for receiving an application description from an author from over a computer communications network, for parsing said application description to identify object parameters for a computing application, to locate a coding module corresponding to at least one of the object parameters within a node, for supplying said application description to said node in which said located coding module applies said application description to generate at least one output object corresponding to the identified object parameters in said application description, and returning said at least one output object to the author over the computer communications

network *(see Figures 1 and 2; Paragraph [0006], "... a file, such as a database mapping description or declaration, is authored by a user or a design tool in a particular data language in which a format can be defined, such as XML. Such an exemplary file is referred to as a blueprint ..."; Paragraph [0017], "Blueprints allow the ASP.NET markup-and-code paradigm to be extended to other domains such as user interfaces, database mapping, web services, and compiled extensible stylesheet language (XSL) transforms."; Paragraph [0035], "Upon receiving the blueprint 200, the blueprint translator 210 parses the blueprint (using, e.g., an XML parser), and provides the parsed blueprint to a Document Object Model (DOM) for further processing. The output of the DOM is provided to a semantic analyzer and code generator. Source code 220 is thereby generated in accordance with predetermined schemas, patterns, and/or hierarchical rules, for example. The source code 220 may access or point to a supporting framework or class library 230."; Paragraph [0047], "In addition, the framework defines a file extension, .dbml, and includes a blueprint translator that can translate .dbml files containing XML-formatted mapping descriptions into source code that targets the framework. An exemplary .dbml blueprint is generated by a user or a design tool and is shown below."; Paragraph [0087], "The present invention can be applied to a wide variety of technologies, such as ... web services ..."; Paragraph [0058], "... a blueprint translator can use the CodeDOM (an object model for abstract syntax trees and code generation provided in the System.CodeDom namespace) to generate source code in a language-neutral fashion.").*

However, Hejlsberg does not disclose:

- a computational grid, wherein said computational grid includes a plurality of computers sharing computational resources, said computational grid comprising a plurality of nodes, each node comprising at least one programming model.

Ferstl discloses:

- a computational grid, wherein said computational grid includes a plurality of computers sharing computational resources, said computational grid comprising a plurality of nodes, each node comprising at least one programming model *(see Column 1: 52-67 to Column 2: 1-8, "A computing grid is a hardware and software infrastructure serving to handle computing jobs submitted by a user. The computing grid may interconnect distributed computers, storage devices, mobile devices, instruments, sensors, data bases and/or software applications. Generally a computing grid may comprise virtually any kind of computing device and includes a grid infrastructure to handle the distribution of computing jobs." and "Upon receiving an instruction to distribute a computing job the grid infrastructure selects a suitable computing device and transfers the computing job to the selected computing device." and "Accordingly, a user or application at a client device may issue an instruction to execute a computing job towards the grid infrastructure which in turn selects a suitable processing element and the processing results are ultimately returned to the client.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Ferstl into the teaching of Hejlsberg to modify Hejlsberg's invention to include a computational grid, wherein said computational grid includes a plurality of computers sharing computational resources, said computational grid comprising a plurality of nodes, each node comprising at least one programming model. The modification

would be obvious because one of ordinary skill in the art would be motivated to utilize a variety of computing devices and/or software applications to quickly solve a single computing task *(see Ferstl – Column 1: 46-50)*.

As per **Claim 21**, the rejection of **Claim 19** is incorporated; and Hejlsberg further discloses:

- wherein said application description is generated using Object Meta Language (OML) *(see Paragraph [0006], "... a file, such as a database mapping description or declaration, is authored by a user or a design tool in a particular data language in which a format can be defined, such as XML. Such an exemplary file is referred to as a blueprint ...")*.

As per **Claim 22**, the rejection of **Claim 21** is incorporated; and Hejlsberg further discloses:

- wherein said OML is an eXtensible Markup Language (XML) dialect *(see Paragraph [0006], "... a file, such as a database mapping description or declaration, is authored by a user or a design tool in a particular data language in which a format can be defined, such as XML. Such an exemplary file is referred to as a blueprint ...")*.

As per **Claim 23**, the rejection of **Claim 19** is incorporated; and Hejlsberg further discloses:

- wherein said coding modules are XML templates *(see Paragraph [0047], "In addition, the framework defines a file extension, .dbml, and includes a blueprint translator that*

can translate .dbml files containing XML-formatted mapping descriptions into source code that targets the framework.").

As per **Claim 24**, the rejection of **Claim 19** is incorporated; and Hejlsberg further discloses:

- wherein said coding modules are eXtensible Stylesheet Language (XSL) style sheets *(see Paragraph [0017], "Blueprints allow the ASP.NET markup-and-code paradigm to be extended to other domains such as user interfaces, database mapping, web services, and compiled extensible stylesheet language (XSL) transforms.").*

As per **Claim 25**, the rejection of **Claim 24** is incorporated; and Hejlsberg further discloses:

- parsing said description to locate at least one variable *(see Paragraph [0048], "... mapping the Customers table in the database to a Customer class in the Northwind namespace. Further details of the mapping include the CustomerID column that maps to an Id property, the ContactName column that maps to a Name property, etc.")*; and

- substituting said at least one variable with at least one replacement variable, wherein said at least one replacement variable is the result of an XML/XSL transform *(see Paragraph [0048], "... the blueprint calls for an Orders collection to be generated in the Customer class based on the relation between the Customer and Order classes described in the <relation> element."; Paragraph [0050], "A blueprint like the one set forth above would typically be*

generated by a database design tool, but it could also be authored manually or created by an XML transformation.").

As per **Claim 26**, the rejection of **Claim 23** is incorporated; and Hejlsberg further discloses:

- parsing said description to locate at least one variable *(see Paragraph [0048], "... mapping the Customers table in the database to a Customer class in the Northwind namespace. Further details of the mapping include the CustomerID column that maps to an Id property, the ContactName column that maps to a Name property, etc.")*; and

- substituting said at least one variable with at least one replacement variable, wherein said at least one replacement variable is stored in said XML template *(see Paragraph [0048], "... the blueprint calls for an Orders collection to be generated in the Customer class based on the relation between the Customer and Order classes described in the <relation> element."; Paragraph [0050], "A blueprint like the one set forth above would typically be generated by a database design tool, but it could also be authored manually or created by an XML transformation.").*

Response to Arguments

11. Applicant's arguments filed on September 28, 2009 have been fully considered, but they are not persuasive.

In the Remarks, Applicant argues:

a) While Examiner is correct that Ferstl clearly discloses a computational grid of nodes, the combination of Ferstl and Hejlsberg (and especially Ferstl) provides no teaching (not even a hint of a teaching) directed to the selection of a coding module in a node that corresponds to identified object parameters in a description of a computing application. To wit, the cited portion of Ferstl--namely column 1, line 65 through column 2, line 8, only describes the receipt of an instruction to distribute a computing job to a selected computing device in a computing grid. There is no mention that a "coding module" in a node (equated by Examiner to be a 'computing device') is selected based upon identified object parameters in a description of a computing application. So much, however is required by the amended claim language of claims 1, 12 and 19.

Examiner's response:

a) Examiner disagrees. With respect to the Applicant's assertion that there is no mention that a "coding module" in a node is selected based upon identified object parameters in a description of a computing application, the Examiner respectfully submits that Ferstl clearly discloses "locating a coding module corresponding to at least one of object parameters within a node contained within a computational grid" *(see Column 1: 52-59, "A computing grid is a hardware and software infrastructure serving to handle computing jobs submitted by a user. The computing grid may interconnect distributed computers, storage devices, mobile devices, instruments, sensors, data bases and/or software applications. Generally a computing grid may comprise virtually any kind of computing device and includes a grid infrastructure to handle the distribution of computing jobs." and 65-67 to Column 2: 1-8, "Upon receiving an instruction to*

distribute a computing job the grid infrastructure selects a suitable computing device and transfers the computing job to the selected computing device." and "Accordingly, a user or application at a client device may issue an instruction to execute a computing job towards the grid infrastructure which in turn selects a suitable processing element and the processing results are ultimately returned to the client."; Column 12: 25-31, "In an example, the selection section obtains the selection information and accesses data specifying corresponding features of a plurality of job handlers, for example, in a configuration file specifying features of a plurality of job handlers available. Then, the selection section may identify a suitable job handler matching the selection information in association with the job request."). Note that Ferstl discloses that a computing grid is a hardware and software infrastructure interconnecting distributed computers, storage devices, mobile devices, instruments, sensors, data bases and/or software applications. Ferstl also discloses that the computing grid selects a suitable processing element to process a computing job. Thus, one of ordinary skill in the art would readily comprehend that the suitable processing element may be a software application that is selected to process the object parameters of the computing job.

Therefore, for at least the reason set forth above, the rejections made under 35 U.S.C. § 103(a) with respect to Claims 1, 12, and 19 are proper and therefore, maintained.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure.

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

14. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Q. C./

Examiner, Art Unit 2191

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191